Q.2 a. Differentiate between non-interactive and interactive graphics.

Answer:

Computer graphics is an art of drawing pictures, lines, charts, etc using computers with the help of programming. Computer graphics is made up of number of pixels. Pixel is the smallest graphical picture or unit represented on the computer screen. Basically there are two types of computer graphics namely.

Interactive Computer Graphics: Interactive Computer Graphics involves a two way communication between computer and user. Here the observer is given some control over the image by providing him with an input device for example the video game controller of the ping pong game. This helps him to signal his request to the computer.

The computer on receiving signals from the input device can modify the displayed picture appropriately. To the user it appears that the picture is changing instantaneously in response to his commands. He can give a series of commands, each one generating a graphical response from the computer. In this way he maintains a conversation, or dialogue, with the computer.

Interactive computer graphics affects our lives in a number of indirect ways. For example, it helps to train the pilots of our airplanes. We can create a flight simulator which may help the pilots to get trained not in a real aircraft but on the grounds at the control of the flight simulator. The flight simulator is a mock up of an aircraft flight deck, containing all the usual controls and surrounded by screens on which we have the projected computer generated views of the terrain visible on take off and landing.

Flight simulators have many advantages over the real aircrafts for training purposes, including fuel savings, safety, and the ability to familiarize the trainee with a large number of the world's airports.

Non Interactive Computer Graphics: In non interactive computer graphics otherwise known as passive computer graphics. it is the computer graphics in which user does not have any kind of control over the image. Image is merely the product of static stored program and will work according to the instructions given in the program linearly. The image is totally under the control of program instructions not under the user. Example: screen savers

b. Describe the display technology used in TFT screen.

Answer:

Thin Film Transistor, flat-panel display screen in which each pixel is controlled by one to four transistors. TFTs are color displays also called active-matrix LCDs.

High color saturation, high contrast and with pixel density >100 ppi. High speed and good viewing angle for these long life-time color displays. It is available in wide range of sizes.

Flat-panel displays (FPDs) are becoming increasingly commonplace in today's commercial electronic devices. FPDs are finding widespread use in many new products, such as cellular phones, personal digital assistants (PDAs), camcorders, and laptop personal computers (PCs). This generation of handheld electronics places stringent demands on their displays. FPDs in these devices are expected to be lightweight, portable, rugged, low-power and high-resolution. Displays having all these attributes will enable a wide variety of commercial applications in the future.

Active-matrix liquid-crystal displays (AMLCDs) are the leading flat-panel display technology. These displays are ubiquitous in laptops, often dubbed "active-matrix TFT," (an abbreviation for "active-matrix thin-film transistor"). What exactly does this name mean?

A display is composed of a grid (or matrix) of picture elements ("pixels"). Thousands or millions of these pixels together create an image on the display. Thin-film transistors (TFTs) act as switches to individually turn each pixel "on" (light) or "off" (dark). The TFTs are the active elements, arranged in a matrix, on the display. Thus, the name "active-matrix TFT."

Most commercially available AMLCDs use glass as the starting material in the display fabrication process. Glass has excellent optical clarity and is compatible with chemicals used in standard semiconductor processing. However, glass has the undesirable characteristic of being extremely fragile. As a result, displays must be handled carefully to avoid breakage. However, if plastic is employed as the starting material for display fabrication, we can achieve a display that is not only lightweight and rugged but also flexible. The realization of such a technology will have a significant impact on the display industry. It is, however, not a trivial task to fabricate displays on plastic. Many significant challenges arise when plastic substrates are used in place of glass. Research in the TFT group is aimed at addressing and overcoming those challenges.

The development of a thin-film transistor (TFT) technology for use with plastic substrates is still in its infancy. There is significant room for improvement in ultra-low temperature fabricated poly-Si TFTs. High mobilities, low leakage currents and threshold voltages are desirable for high-performance active-matrix LCD applications, particularly for the integration of driver circuitry, but low processing temperatures (<150ºC) must be maintained for compatibility with low-cost plastic substrate materials. In general, superior poly-Si TFT performance is achieved with higher-temperature fabrication processes, because the quality of the critical gate-dielectric interface is highly sensitive to process temperature.

An ultra-low-temperature (100ºC) fabrication process, which would be compatible with flexible plastic substrates, is being developed by the TFT technology group at UCB. The goal is to achieve polycrystalline-silicon (poly-Si) thin-film transistors (TFTs) with current-driving capability far exceeding that of conventional amorphous-silicon TFTs typically employed in high-performance active-matrix liquid-crystal displays today. Techniques for formation of the poly-Si and gate-dielectric materials are being investigated in order to determine the optimum processes for high-performance TFTs. Various device and process architectures for attaining low leakage current are being studied. The degradation of TFT performance under high-voltage bias stressing will also be characterized. To enable device modeling and circuit design, a physically based model for ultra-low-temperature-fabricated TFTs will be developed.

Common Applications Widely used for mobile phones displays and computer screens:
Medical, communication, industrial, marine and office equipments.

A transistor whose active, current-carrying layer is a thin film (usually a film of silicon), in contrast to MOSFETs, which are made on Si wafers and use the bulk-silicon as the active layer. In a flat-panel display, light must be able to pass through the substrate material to reach the viewer. Opaque silicon wafers obviously will not be suitable for these transmissive displays. Glass is the most commonly used starting substrate because it is highly transparent and is compatible with conventional semiconductor processing steps. Since glass is not a semiconductor like silicon, a thin film of silicon is deposited on top and the transistors are fabricated using this thin layer. Hence, the name "thin-film transistor."

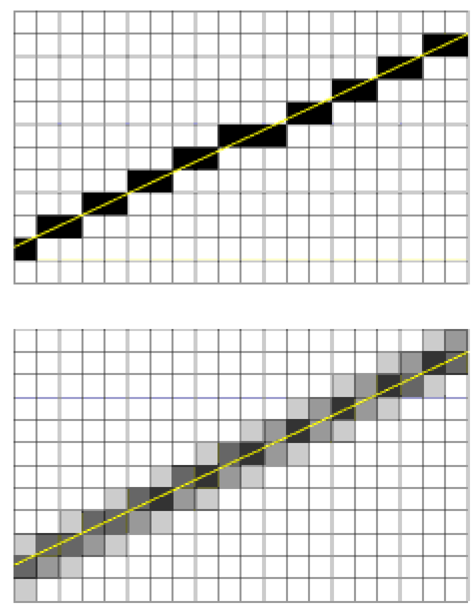
Q.3 a. Why only 1/8th of the total number of pixels on the circumference of a circle is required to be computed? Write an algorithm to draw a circle.

Answer: Refer to the prescribed text book.

b. What is aliasing? Write a method to resolve the aliasing effect while printing a graphics on printer or drawing the graphics on screen.

Answer:

When a line in a digital image runs at an angle, then it will often appear with jagged edges. This effect is caused by the regular pixel grid in the image, and is called **aliasing**.



Example of aliased and anti-aliased line

To avoid this effect, the process of **anti-aliasing** paints some nearby pixels in an intermediate color or brightness. That way, the visual appearance of the line (or an edge) is smoothed out.

The problem of **aliasing** is prevalent in synthetic images created by methods such as *ray tracing*. Some programs try to circumvent it by sending more than one ray per pixel (oversampling), and interpolating their result. With the *Radiance software*, the same result is reached by rendering an image that is two or three times as big as the target size with rpict(1), and then *filtering* this intermediate image down to scale with pfilt(1). To get even smoother results, the image can also be rendered with *jittered sampling*.

Q.4 a. Write the 2D transformation matrices for three basic transformations: translation, rotation and scaling.

Answer: Refer to the prescribed text book.

b. Compute the transformation matrix for reflecting a point (x, y) with respect to line y = x and hence find the new coordinates for the point P(2, -4).

Answer:

Rotate P(2, -4) in clockwise direction by 45^0 and then find reflection of P(2, -4) with respect to x- axis and then rotate the point again by 45^0 in anti-clockwise direction.

Q.5 a. What is Bezier Surface? How is it related to Bezier curve?

Answer:

A surface defined by mathematical formulae, used in computer graphics. A surface P(u, v), where u and v vary orthogonally from 0 to 1 from one edge of the surface to the other, is defined by a set of (n+1)*(m+1) "control points" (X(i, j), Y(i, j), Z(i, j))

for i = 0 to n, j = 0 to m.
P(u, v) = Sum i=0..n Sum j=0..m [(X(i, j), Y(i, j), Z(i, j)) * B(i, n, u) * B(j, m, v)]
B(i, n, u) = C(n, i) * u^i * (1-u)^(n-i)
C(n, i) = n!/i!/(n-i)!

Bezier surfaces are an extension of the idea of Bezier curves, and share many of their properties.

he Bézier surface is formed as the cartesian product of the blending functions of two orthogonal Bézier curves.

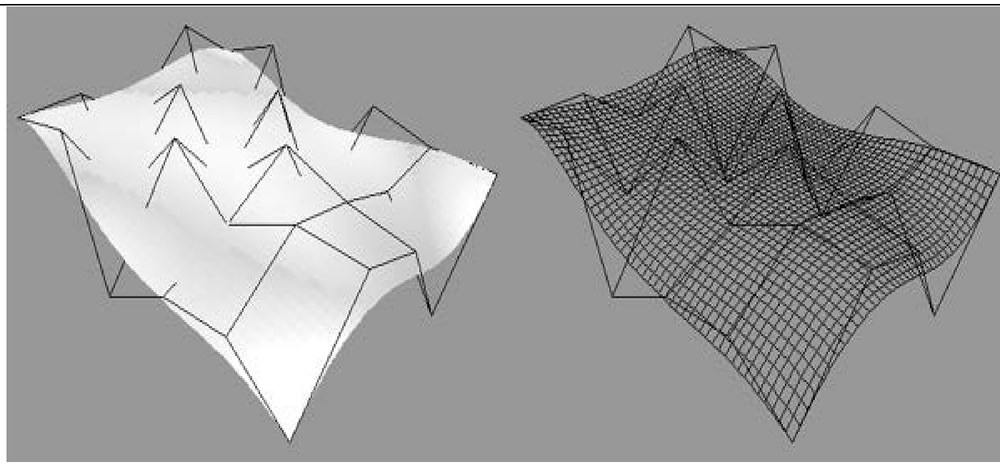
$$\mathbf{B}(u,v) = \sum_{i=0}^{Ni} \sum_{j=0}^{Nj} P_{i,j} \frac{Ni!}{i!\,(Ni-i)!} u^i (1-u)^{Ni-i} \frac{Nj!}{j!\,(Nj-j)!} v^j (1-v)^{Nj-j}$$

$$0 \le u \le 1$$
$$0 \le v \le 1$$

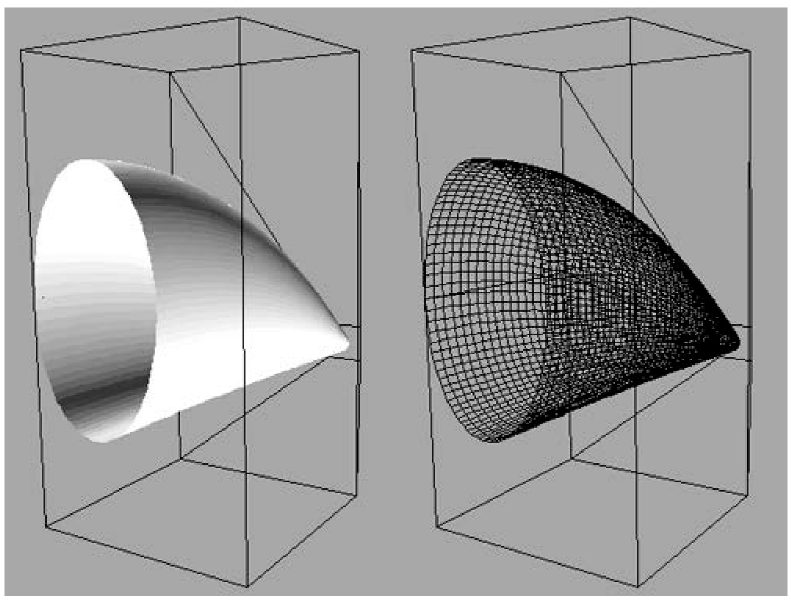
Where $P_{i,j}$ is the i,jth control point. There are N_{i+1} and N_{j+1} control points in the i and j directions respectively.

The corresponding properties of the Bézier curve apply to the Bézier surface.
- The surface does not in general pass through the control points except for the corners of the control point grid.
- The surface is contained within the convex hull of the control points.

Along the edges of the grid patch the Bézier surface matches that of a Bézier curve through the control points along that edge.



Closed surfaces can be formed by setting the last control point equal to the first. If the tangents also match between the first two and last two control points then the closed surface will have first order continuity.



While a cylinder/cone can be formed from a Bézier surface, it is not possible to form a sphere.

b. Briefly explain the concept of perspective projection and provide a situation showing three vanishing points.

Answer: Refer to the prescribed text book.

Q.6 a. Find a transformation matrix for converting an image of size (W, H) = (300, 250) to display it on a viewport of size (W*, H*) = (200, 150). Lower left corner of the view port is at (200, 200).

Answer: Refer to the view port transformation in the prescribed book.

b. Write the Cohen Sutherland line clipping algorithm.

Answer: Refer to the prescribed text book.

Q.7 a. Write Z- Buffer algorithm for hidden surface removal.

Answer:

Perspective transformation maps viewing pyramid to viewing box in a manner that maps lines to lines. This transformation also maps polygons to polygons.
Idea: When we scan convert, step in *z* as well as *x* and *y*.
In addition to framebuffer, we'll have a depth buffer (or *z* buffer) where we write *z* values
Initially, *z* buffer values set to ∞

Depth of far clipping plane (usually 1) will also suffice

Scan convert using the following WritePixel:

```
WritePixel(int x, int y, float z, colour)
if ( z < zbuf[x][y] ) then
    zbuf[x][y] = z;
    frambuffer[x][y] = colour;
end
```

b. Write an algorithm to remove hidden lines from the scene before drawing it on the display screen.

Answer:

Write any algorithm to remove ***hidden lines*** from scenery, and store the image in frame buffer before rendering it on screen.
When rendering a three dimensional surface on a two dimensional screen, lines which cannot be seen by the viewer must be removed from the screen so that shape of the surface is not distorting by overlapping lines. In order to remove these lines, hidden line algorithms are applied in the surface rendering software to create a wireframe which contains only visible lines and hides the lines covered by the surface.

An algorithm which is often used is the one developed by Arthur Appel at IBM in the late 1960′s. “This algorithm works by propagating the visibility from a segment with a known visibility to a segment whose visibility is yet to be determined”. By a comparison of the two following images, the line removal algorithm can be seen at work as the wireframe representation of the surface shaded object removes the lines which are not in view.

Although the ability of the algorithm to correctly remove hidden lines is clear, the way in which the algorithm removes these lines is hard to see. The basic algorithm is based on the principle stated by Walter Hedger in his paper on Vector Hidden Line Removal and Fractional Quantitative Invisibility.

"The topological silhouettes of a model are those edges that bound front-facing and back-facing regions of faces. Collectively, these edges are usually known as the silhouettes of a model with respect to a particular view vector. First the algorithm starts by projecting all of the edges into curves … classified as either back-facing, front-facing, or silhouettes … until the classification is unambiguous for each curve in its interior."

From this point, the three dimensional rendering of the function can be formed by using the quantitative invisibility (q.i.) which represents the number of faces which obscure a point. For Appel's algorithm a count of the number of obscuring front-facing faces (q.i.) on all the projected curves is used to implement hidden line removal.

Although much of the initial work in the field of hidden line removal was done by Arthur Appel, the field is still growing as there are exceptions when his algorithm is not effective. There exist a variety of other algorithms which are implemented in computer-assisted design (CAD) such as the object-precision algorithms of Weiss and Galimberti/Montenari and the image-precision algorithms Encarnacao (priority-edge intersection test and scan grid – point/surface test), Warnock, and Watkins.

The most popular algorithms which are being used and enhanced are z-buffering and face drawing algorithms like binary space partitioning. Z-buffering compares all the interior points of faces and therefore is much slower. Face drawing is faster since the hidden edges are overwritten, but computing the points in the interior of the face is time consuming. Furthermore, floating point computations are used to make binary decisions which can cause problems when the decision is wrong.

Although programs are currently limited to utilizing algorithms with the least error, the development of the original algorithms of Arthur Appel has opened up a new field of vector hidden line removal as people begin to search for fast, effective, and error-free algorithm which can be used for hidden line removal.

Q.8 a. Write any four video formats and explain any one of them.

Answer:

MPEG (Moving Pictures Expert Group): three video formats, MPEG 1, 2, and 4.
MPEG---1: Old, supported by everything (at least up to 352x240), reasonably efficient. A good format for the web.

MPEG---2: A version of MPEG---1, with better compression. 720x480. Used in HDTV, DVD, and SVCD.

MPEG---4: A family of codecs, some of which are open, others Microsoft proprietary.

H.264: Most commonly Used codecs for videos uploaded to the web. Part Of the MPEG---4 codec.

MPEG spinoffs: mp3 (for music) and VideoCD.

MJPEG (Motion JPEG): A Codec consisting of a stream of JPEG images. Common In video from digital cameras, and a reasonable format for editing videos, but it doesn't compress well, so it's not good for web distribution.

DV (Digital Video): Usually Used for video grabbed via firewire off a video camera. Fixed at 720x480 @ 29.97FPS, Or 720x576 @ 25 FPS. Not very highly compressed.

WMV (Windows Media Video): A Collection of Microsoft Proprietary video codecs. Since Version 7, It has used a special version of MPEG4.

RM (Real Media): a closed codec developed by Real Networks for streaming video and audio.

DivX: In early versions, essentially an ASF (incomplete Early MPEG---4) Codec inside an AVI container; DivX 4 And later are a more full MPEG---4 codec…no resolution limit. Requires More horsepower to play than mpeg1, but less than mpeg2. Hard to find mac and windows players.

Sorenson 3: Apple's Proprietary codec, commonly used for distributing movie trailers (inside a Quicktime container).

Quicktime 6: Apple's Implementation of an MPEG4 codec.

RP9: A very efficient streaming proprietary codec from Real (not MPEG4).

WMV9: A proprietary, non---MPEG4 codec from Microsoft.

Ogg Theora: A Relatively new open format from Xiph.org.

Dirac: A Very new open format under development by the BBC.

b. What is real time animation and how is it produced?

Answer:

Real-Time Animation, sometimes called Machinima (muh-sheen-eh-mah) is filmmaking within a real-time, 3D virtual environment, often using 3D video-game technologies. In an expanded definition, it is the convergence of filmmaking, animation and game development. Real-Time Animation is real-world filmmaking techniques applied within an interactive virtual space where characters and events can be either controlled by humans, scripts or artificial intelligence.

By combining the techniques of filmmaking, animation production and the technology of real-time 3D game engines, Real-Time Animation makes for a very cost- and time-efficient way to produce films, with a large amount of creative control.

Real-Time Animation can be produced in a couple of ways. It can be script-driven, whereas the cameras, characters, effects etc. are scripted for playback in real-time. While similar to animation, the scripting is driven by events rather than keyframes. It can also be recorded in real-time within the virtual environment, much like filmmaking (the majority of game-specific Real-Time Animation pieces are produced in this fashion). While both of these approaches have their pros and cons, they are both Real-Time Animation-making techniques.

Real-Time Animation provides:

- the real-time recording of human/scripted performances and events – akin to shooting film, eliminates the rendering process.
- the creative flexibility of artistic assets, allows total control over visual representation of characters, events, etc.
- an interactive environment provides a space where characters can interact and real-world physics can be reproduced.
- Hardware driven playback is resolution independent.

Because Real-Time Animation can be shot live or scripted in real-time, it's much faster to produce than traditional CGI animation. A live action director can feel right at home and an animation director will be able to direct without having to rely on key frames. Multiple takes can be made in real-time or just a few takes while the rest is adjusted in post, dependent on the director's style.

Q.9 a. What are the various components of multimedia? How do they affect human perception and understanding?

Answer: Refer Page Nos 443 from Text Book

b. What are the differences between BMP and PCX file format?

Answer: Refer Page Nos 447-449 from Text Book

<u>TEXT BOOK</u>

I. Computer Graphics, Amarendra N. Sinha, Arun D Udai, TMH, 2008.